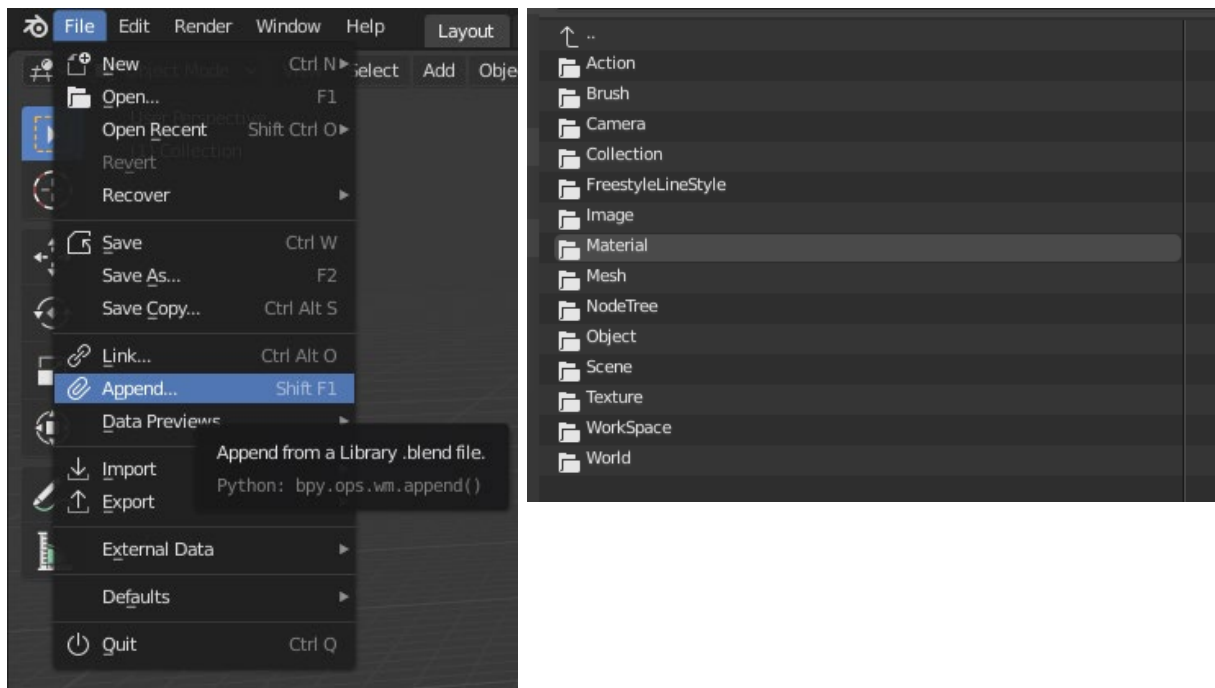


Dermis Shader Documentation

Dermis is a multi-level SSS shader for Blender, inspired by Anders Langland's multi-level SSS shader for Arnold 4. It affords the user more nuanced and intuitive control over skin tones and a more natural look scattering colour and depth. The shader of course gives the nicest looking and most accurate results in Cycles, but it can work for Eevee too - however Eevee can only use one SSS shader at a time, so while Dermis can be used to get finer colour control for your SSS, it's very much overkill as it will only use the SSS lobe of a principled BSDF shader, with the colour inputs and control for the first (upper) layer of Dermis.

How To Import the Material into your Scene

Open the destination scene in Blender and navigate to the file drop down. Select Append, and in the file browser navigate to the .blend file you downloaded from Gumroad. Navigate to the folder within the .blend file called "Materials" and select the material called "Dermis Shader", then click append in the top right of the window.

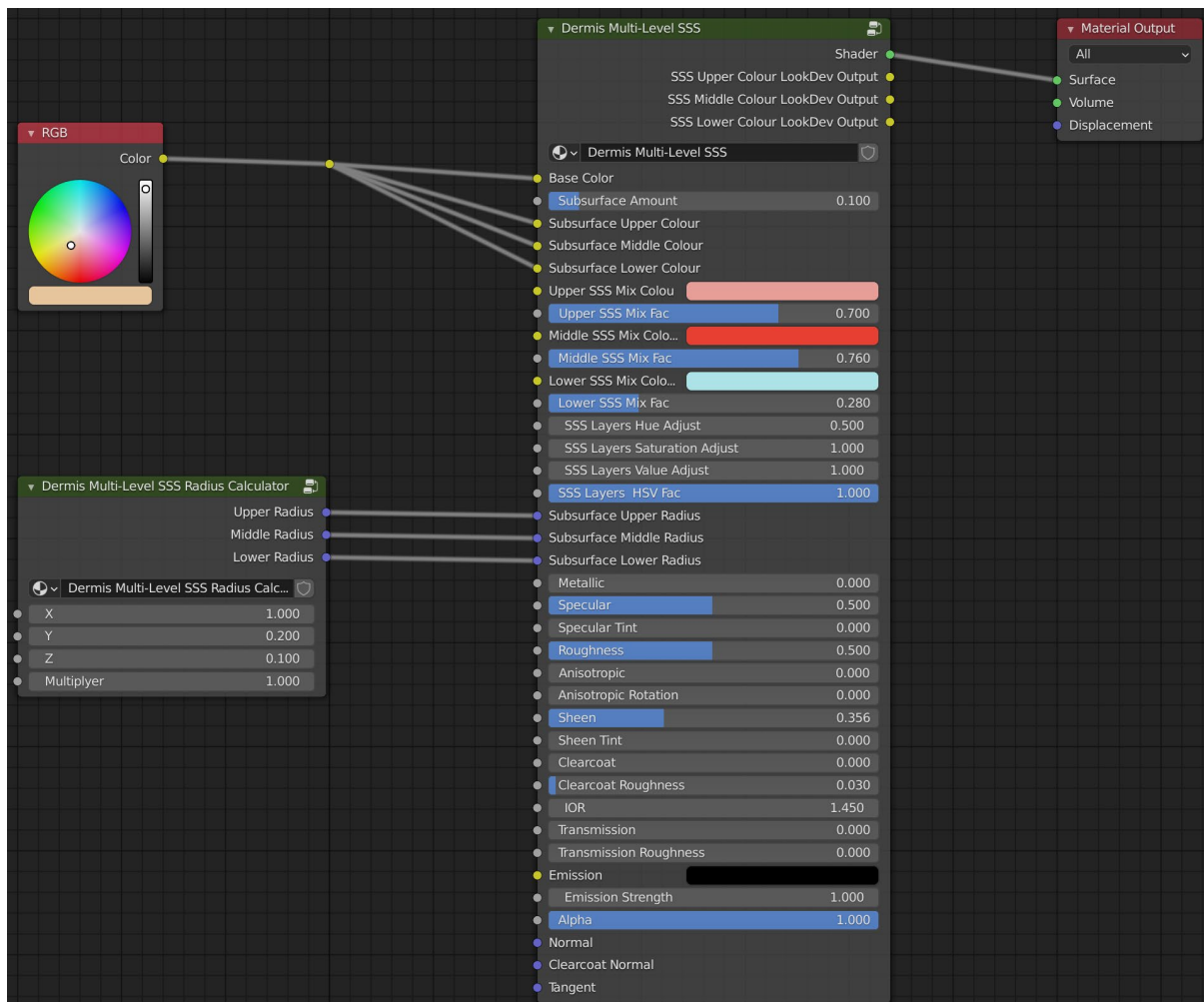


How to use the Dermis Shader

The Dermis shader offers all the usual controls of a Cycles Principled BSDF Shader node, but with three times as many controls for Sub-Surface Scattering, along with specific colour correction controls to enable finer skin tone tweaking. These controls are present in triplicate, and work in parallel to the Principled node's original SSS component. As such, for more intuitive control, an additional node group, the Multi-Level Radius Calculator, is provided. This node enables the use to enter the usual sub-surface radius vector values one would enter for a normal principled node, along with a multiplier. This multiplier can be considered a second control for the "depth" of the scattering (along with the Subsurface

Amount slider), and it also controls the difference in radius between each SSS layer, controlling much each layer transmits each light channel (R =X, G = Y, B = Z)

The recommended setup for the shader is as follows, for original Cycles:



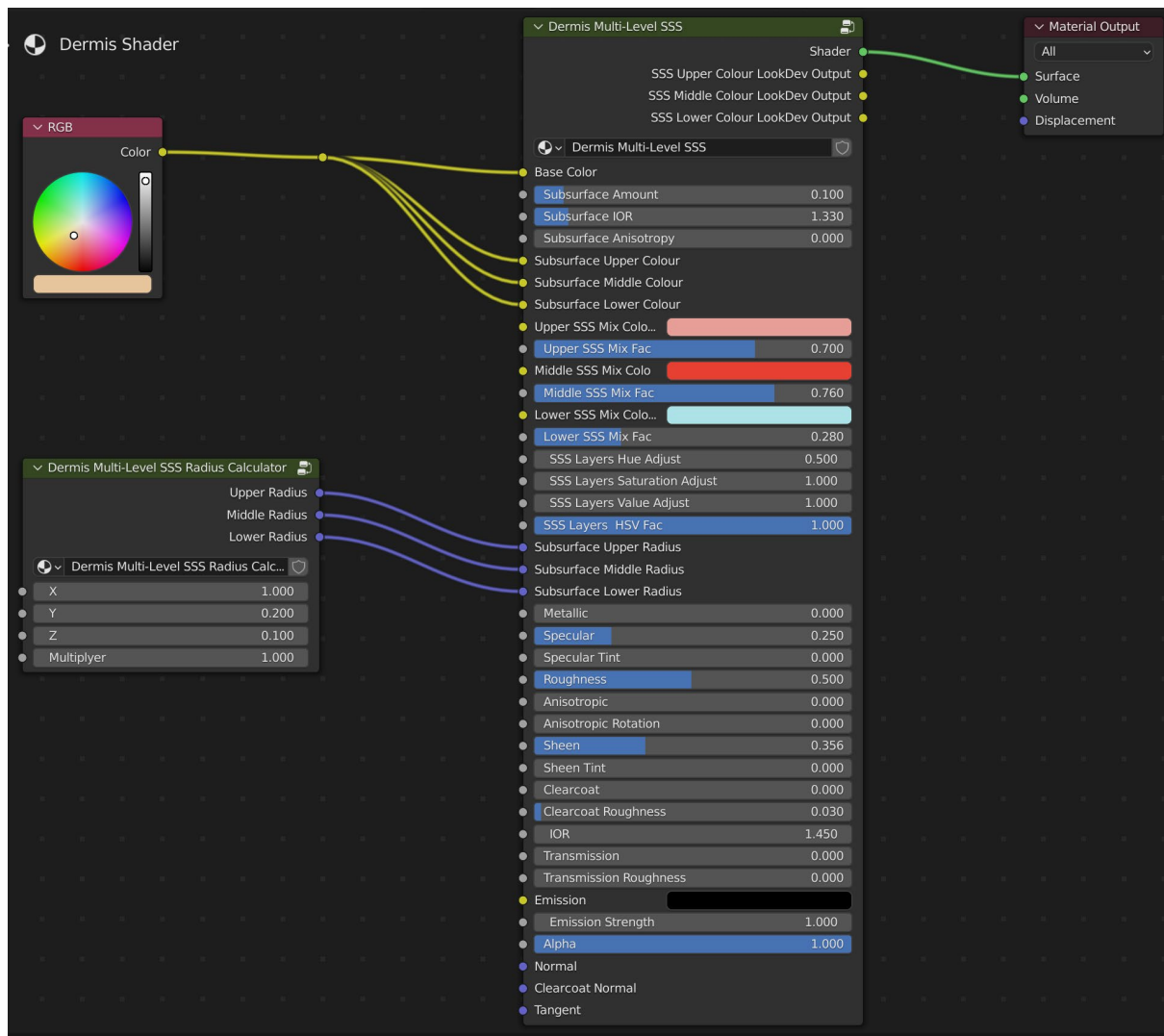
For the simplest setup with PBR textures, connect everything as normal, but plug your albedo/diffuse texture into the reroute node (the yellow dot) that the RGB node is plugged into. If you have maps with a different SSS colour map, connect your albedo/diffuse to the Base Colour input, and connect your SSS map to the three SSS colour inputs using the same reroute node. If you have multiple layers of SSS colour maps, do something similar, but with the layer maps connect to their corresponding inputs (upper, middle, lower colour inputs).

The colour channels are driven by mix shaders that blend the approximate skin tones for each skin layer (provided by the colour ramp nodes) with either a standard skin albedo texture, or a standard single-layer SSS colour texture. Three HSV adjustment nodes are built into the shader, and these controls are exposed and repeated to the colour inputs for each SSS layer. These are very useful for tweaking the skin tones of your character, in particular the Saturation slider to make your skin more vibrant, and the value slider to adjust your characters skin tone. If you have a character who's skin looks too pale or too dark, try adjusting these sliders. It's of particular note that in real life, Melanin, the only pigment in skin, simply absorbs light – light skin tones and dark skin tones are actually the same colour,

just very different in terms of how much light they absorb. Characters with particularly dark skin will usually need the value slider to be used to make the SSS input colours darker, as this reduces how far light scatters into the skin realistically, similarly to melanin.

Additionally, in the setup that the shader is supplied in, with no textures used, it should provide a decent light skin toned no-texture skin shader for stylised characters, although you will probably want to tweak the RGB node's colour, perhaps along with the SSS colours/mix factors to achieve the desired skin tone. If you're using a scan that has a scattering depth map like the WikiHuman Emily scan (the single scatter map) you should use it to drive the multiplier of the radius calculator, though I recommend using a colour ramp node so you can tweak it to look just right (if you can't get that to work, try using it to drive the mix nodes of the skin tone colour ramps in conjunction with some math/colour ramp nodes). If you would like to see my setup for Wikihuman's Emily with Dermis, drop me an email or message on the blender market and I will send you the blend file (demo here: <https://www.instagram.com/p/CBS3hPAJTou/>).

And for Cycles X:



The major differences between the Cycles and Cycles X version of the shader are the introduction of three things: The SSS IOR input, the SSS Anisotropy input, and a new diffusion function, Random Walk (Fixed Radius). The SSS IOR input is quite self-explanatory, representing the amount that light rays are refracted when scattering through an object. For human skin this should usually be roughly 1.33, as this is the IOR of water, which of course animal cells (atleast those in organisms from earth!) are made of. The anisotropy input is slightly less intuitive. It controls how directional the subsurface scattering is. When it is low, or 0, light rays will be scattered equally in all directions by the object, subject to the radius vector(s). When it is high or 1, then light rays will be scattered mostly in one direction. The documentation for this effect is currently lacking, so until we get more information, we must assume this direction is “forward”, meaning the light rays will mostly be scattered in the same direction they came in. Finally, the new Random Walk mode. The fixed radius mode of random walk uses the existing random walk diffusion method, but with a fixed radius, and is intended for thin, curved objects, useful for things like leaves and plastic, or ears. The normal random walk mode uses a radius calculated based on the colour, anisotropy, and IOR inputs, although exactly how Cycles calculates this isn’t clear. Here is how the Blender docs put it:

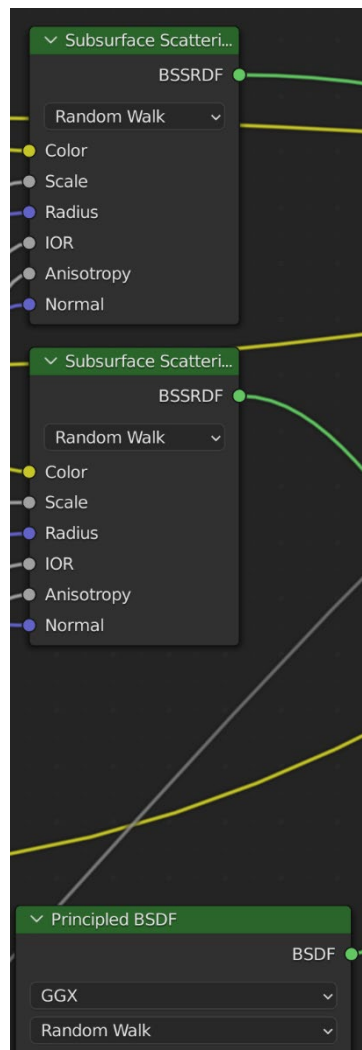
Note

Eevee does not support the *Random Walk* methods.

Christensen-Burley:	An approximation to physically-based volume scattering. This method is less accurate than <i>Random Walk</i> however, in some situations this method will resolve noise faster.
Random Walk (Fixed Radius):	Provides accurate results for thin and curved objects. Random Walk uses true volumetric scattering inside the mesh, which means that it works best for closed meshes. Overlapping faces and holes in the mesh can cause problems.
Random Walk:	Behaves similarly to <i>Random Walk (Fixed Radius)</i> but modulates the <i>Radius</i> based on the <i>Color</i> , <i>Anisotropy</i> , and <i>IOR</i> . This method thereby attempts to retain greater surface detail and color than <i>Random Walk (Fixed Radius)</i> .

Skin is pretty isotropic when it comes to SSS, so this is 0 by default, but play with it! Have some fun making some weird skin, maybe for monsters – things like milk, orange juice, opals have anisotropic scattering, so use them for image reference when playing with anisotropy.

To change what mode the shader is using, since you cant currently control drop-down controls with node groups, select Dermis, press the TAB key, and change the drop downs of the SSS nodes and Principled node to the desired mode:



And remember, only Cycles supports the IOR and Anisotropy inputs, as well as Random Walk. Cycles really is best for this shader.

Dermis SSS Lobe Material

The SSS Lobe material is a mini version of Dermis that is just the SSS lobes you would usually get, with no other parts of the material that you get from the Principled node present. It contains all the same SSS controls as regular Dermis, but none of the PBR controls (since it is just an SSS shader, nothing else), so the same rules outline above apply, minus your PBR workflow. This material is included so that you can easily incorporate Dermis' SSS into your own custom node groups/materials, without any of the PBR stuff you might not want.

Please note that by default the skin shader is using the random walk SSS method, which while the most accurate, can appear quite dark with some HDRs/lighting setups. This may require you to either increase the brightness of your scene lighting, switch to Branched Path Tracing mode (Original Cycles and CUDA/CPU/OpenCL only) in the render settings, or switch the node to Christensen-Burley mode, which you can do by selecting the node and pressing tab to edit the node group, and then changing the scattering method in the drop-down menu for each SSS component, as explained above.

Cheat Sheet: where to plug in what kind of texture

Any nodes that aren't mentioned should be used as normal for a Principled BSDF node.

Node Input	Texture Type
Base Colour	Albedo/Diffuse/Colour Texture
Subsurface Amount	SSS Weight Texture or Manual Setting
Subsurface Upper/Middle/Lower Colour	Appropriate texture for each layer, or Manual Setting, or use the provided Mix Nodes and Colour Ramps.
Subsurface Upper/Middle/Lower Radius	Multi-Level SSS Calculator or Manual Setting (Calculator is recommended)
Metalness/Roughness/Specular/Normal	Connect these to the same inputs you would normally, in the same manner as you would normally, when working with PBR maps and the Principled BSDF shader.

If you make anything using these node groups, I'd appreciate it if you could tag me in an instagram post (@neoncreeper) or email me (nathanaeltownsend@googlemail.com) to let me know. We also have a hashtag, #dermisshader, that you can use on Instagram, Twitter, Facebook etc to tag posts made using Dermis.